

# Fundamentos de Algoritmos e Estrutura de Dados – Aula 03 – Hashing

Prof. André Gustavo Hochuli

[gustavo.hochuli@pucpr.br](mailto:gustavo.hochuli@pucpr.br)

[aghochuli@ppgia.pucpr.br](mailto:aghochuli@ppgia.pucpr.br)

# Plano de Aula

- **Revisão Aula 02**
  - **Inserção e Remoção em Listas**
  - **Lista Duplamente Encadeada**
- **Tabela Hash**
  - **Problema**
  - **Funções Hash**
  - **Colisões**
  - **Trabalho**

# Problema

- Buscar um elemento em tempo constante independente da chave

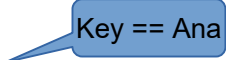
Família	1	2	3	4	5	6
	José Maria	Leila	Artur	Jolinda	Gisela	Alciene

- Imagine um problema para armazenar identificadores de 11 dígitos
  - $10^{11} = 100.000.000.000$  (100bi)
- A busca é custosa
- Aplicar métodos de ordenação a cada 'evento' é custoso

# Dicionário de Dados

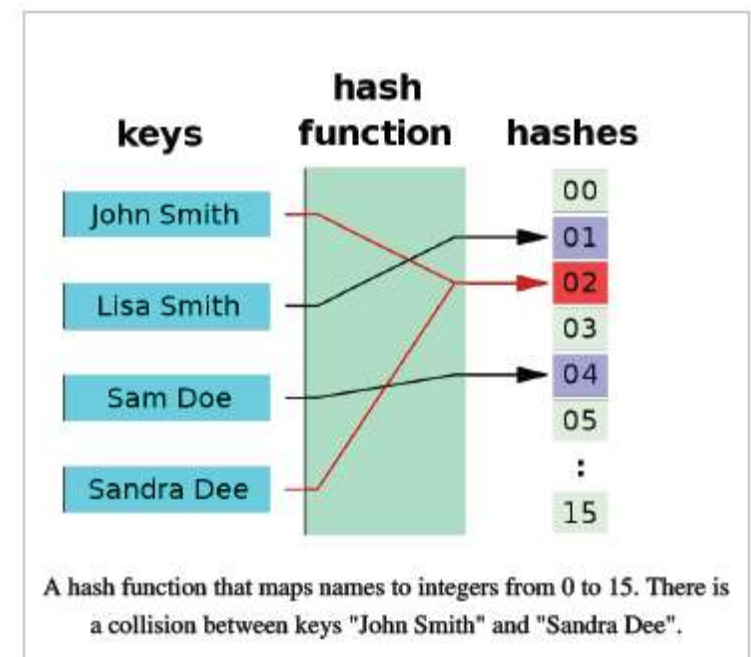
- Tipo de dados abstrato que representa um objeto/entidade
- Implementa as funções Inserir, Buscar e Remover
- Utiliza chaves para indexar a informação (função hash)

```
contatos_lista = [('Yan', '1234-5678'), ('Pedro', '9999-9999'),  
                  ('Ana', '8765-4321'), ('Marina', '8877-7788')]  
  
contatos = dict(contatos_lista)  
print(contatos)  
  
{'Yan': '1234-5678', 'Pedro': '9999-9999', 'Ana': '8765-4321',  
  'Marina': '8877-7788'}  
  
print(contatos['Ana'])  
  
8765-4321
```



# Função Hash e Tabela Hash

- Funções Hash ou Funções de Espalhamento é uma função de mapeamento dos dados para outro domínio
- Não permite caminho inverso (reconstrução)
- A colisão é um fator importante



# Funções Hash - $h(k)$

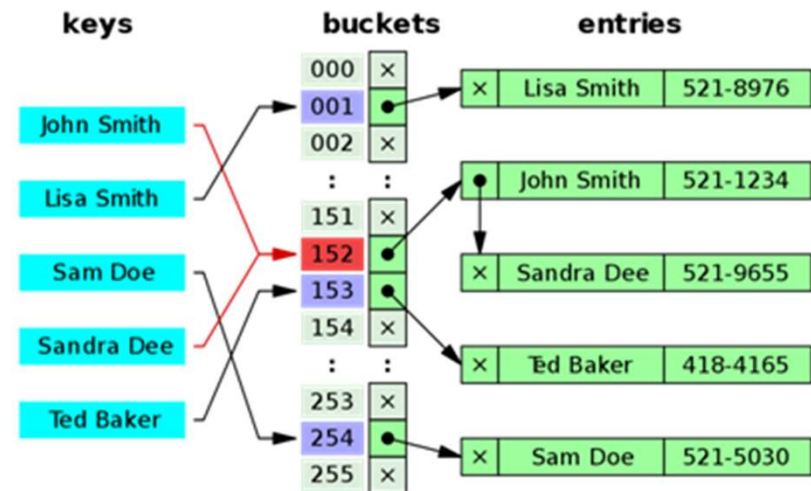
- Módulo
  - Valor de M é crítico (M colisões)

```
int hashCode(int k){  
    return (k % m);  
}
```

- Outros métodos

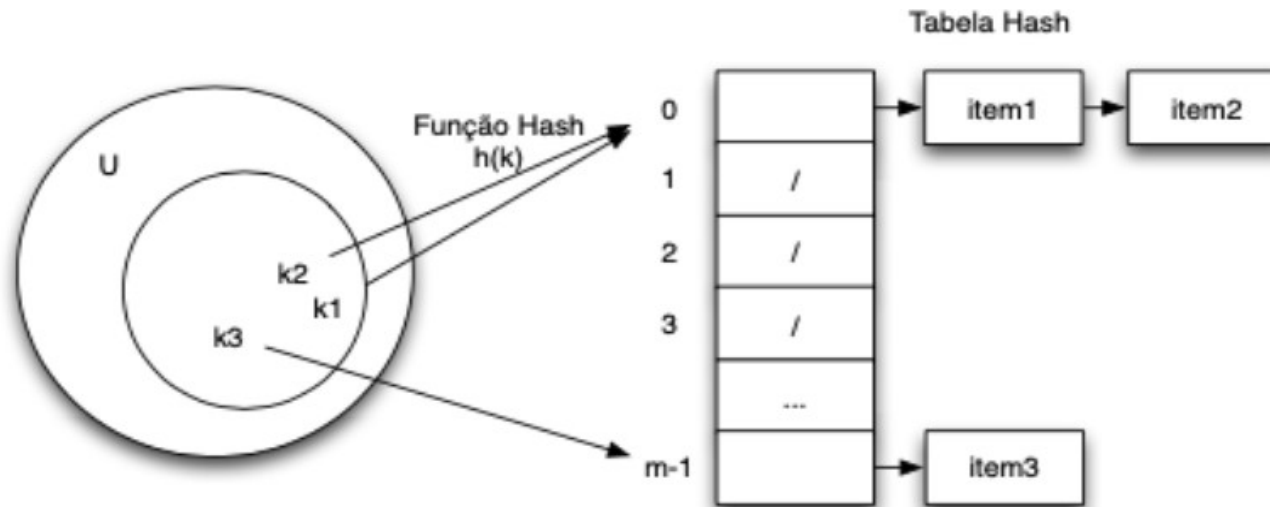
- Multiplicação
- Fibonacci
- Etc...

- Qualidade da hash determina:
  - Colisões vs Custo computacional



# Colisão

- Um ou mais itens mapeados para a mesma chave
- Solução\*: Encadeamento
- Tamanho da lista (#colisões) depende de  $h(x)$



- \*Existem outras soluções disponíveis no estado da arte

# Trabalho

- Implementar uma tabela hash
- Avaliar diferentes funções e seus parâmetros
  - Modular
- Analises Críticas e Comparações
- Formalização no AVA